

MAVEN SNAPSHOTS

http://www.tutorialspoint.com/maven/maven_snapshots.htm

Copyright © tutorialspoint.com

A large software application generally consists of multiple modules and it is common scenario where multiple teams are working on different modules of same application. For example consider a team is working on the front end of the application as app-ui project (app-ui.jar:1.0) and they are using data-service project (data-service.jar:1.0).

Now it may happen that team working on data-service is undergoing bug fixing or enhancements at rapid pace and the they are releasing the library to remote repository almost every other day.

Now if data-service team uploads a new version every other day then following problem will arise

- data-service team should tell app-ui team every time when they have released an updated code.
- app-ui team required to update their pom.xml regularly to get the updated version

To handle such kind of situation, **SNAPSHOT** concept comes into play.

What is SNAPSHOT?

SNAPSHOT is a special version that indicates a current development copy. Unlike regular versions, Maven checks for a new SNAPSHOT version in a remote repository for every build.

Now data-service team will release SNAPSHOT of its updated code everytime to repository say data-service:1.0-SNAPSHOT replacing a older SNAPSHOT jar.

Snapshot vs Version

In case of Version, if Maven once downloaded the mentioned version say data-service:1.0, it will never try to download a newer 1.0 available in repository. To download the updated code, data-service version is be upgraded to 1.1.

In case of SNAPSHOT, Maven will automatically fetch the latest SNAPSHOT (data-service:1.0-SNAPSHOT) everytime app-ui team build their project.

app-ui pom.xml

app-ui project is using 1.0-SNAPSHOT of data-service

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>app-ui</groupId>
  <artifactId>app-ui</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <name>health</name>
  <url>http://maven.apache.org</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>data-service</groupId>
      <artifactId>data-service</artifactId>
      <version>1.0-SNAPSHOT</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

```
</dependencies>
</project>
```

data-service pom.xml

data-service project is releasing 1.0-SNAPSHOT for every minor change

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>data-service</groupId>
  <artifactId>data-service</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>health</name>
  <url>http://maven.apache.org</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
</project>
```

Although, In case of SNAPSHOT, Maven automatically fetches the latest SNAPSHOT on daily basis. You can force maven to download latest snapshot build using -U switch to any maven command.

```
mvn clean package -U
```

Let's open command console, go the **C:\ > MVN > app-ui** directory and execute the following **mvn** command.

```
C:\MVN\app-ui>mvn clean package -U
```

Maven will start building the project after downloading latest SNAPSHOT of data-service.

```
[INFO] Scanning for projects...
[INFO] -----
[INFO] Building consumerBanking
[INFO]    task-segment: [clean, package]
[INFO] -----
[INFO] Downloading data-service:1.0-SNAPSHOT
[INFO] 290K downloaded.
[INFO] [clean:clean {execution: default-clean}]
[INFO] Deleting directory C:\MVN\app-ui\target
[INFO] [resources:resources {execution: default-resources}]
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\MVN\app-ui\src\main\
resources
[INFO] [compiler:compile {execution: default-compile}]
[INFO] Compiling 1 source file to C:\MVN\app-ui\target\classes
[INFO] [resources:testResources {execution: default-testResources}]
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\MVN\app-ui\src\test\
resources
[INFO] [compiler:testCompile {execution: default-testCompile}]
[INFO] Compiling 1 source file to C:\MVN\app-ui\target\test-classes
[INFO] [surefire:test {execution: default-test}]
[INFO] Surefire report directory: C:\MVN\app-ui\target\
surefire-reports
-----
T E S T S
-----
Running com.companyname.bank.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.027 sec

Results :
```

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] [jar:jar {execution: default-jar}]

[INFO] Building jar: C:\MVN\app-ui\target\
app-ui-1.0-SNAPSHOT.jar

[INFO] -----

[INFO] BUILD SUCCESSFUL

[INFO] -----

[INFO] Total time: 2 seconds

[INFO] Finished at: Tue Jul 10 16:52:18 IST 2012

[INFO] Final Memory: 16M/89M

[INFO] -----